

**THE CLAIMS**

1. A method comprising:  
executing a first instance of a program on a first processor in a computer system having multiple processors, and wherein the program refers to a virtual memory address (VMA) in a page table to obtain a pointer to a memory location to write writable data;  
executing a second instance of the program on a second processor in the computer system, and wherein the second instance of the program refers to a virtual memory address (VMA) in a page table to obtain a pointer to a memory location to write the writable data; and  
wherein the VMA referred to by each of the first and second instance of the program is the same, and wherein the VMA referred to by the first instance of the program points to a memory coupled to the first processor, and wherein the VMA referred to by the second instance of the program points to a memory coupled to the second processor.
2. The method as defined in claim 1 further comprising:  
wherein the executing the first instance step further comprises executing the first instance of the program in a first functional unit of the multiple processor system;  
wherein the executing the second instance step further comprises executing the second instance of the program in a second functional unit of the multiple processor system; and  
wherein the first and second instances of the program are replicated versions of the same program.
3. The method as defined in claim 1 wherein the program is an operating system program, and wherein the writable data further comprises a performance counter count value.

4. The method as defined in claim 3 further comprising:  
reading the count value from the memory coupled to the first processor;  
reading the count value from the memory coupled to the second processor; and  
combining the count values.

5. The method as defined in claim 4 wherein the performance counter count value is a number representing a number of page allocations in memory.

6. The method as defined in claim 4 wherein the performance counter count value is a number representing a number of disk accesses.

7. The method as defined in claim 1 wherein the program is an operating system program, and wherein the writable data further comprises a look-aside list header for process control blocks.

8. A computer readable media storing programs executable by a processor that, when executed, perform the following steps:  
accessing a read/write variable in a computer system having a plurality of functional units, each of the plurality of functional units having a processor and a random access memory (RAM) coupled to the processor; the accessing by referring to a virtual memory address (VMA) in a page table to locate the read/write variable, wherein the VMA in each functional unit is the same, and wherein the VMA in each functional unit contains a pointer to RAM within its functional unit.

9. The computer readable media as defined in claim 8 wherein the steps performed by the programs further comprise:

reading each of the read/write variables throughout the computer system;  
combining the read/write variables; and  
writing the combined read/write variables to a single location within the computer system.

10. The computer readable media as defined in claim 9 wherein the combining step further comprises adding the values of each of the read/write variables.
11. The computer readable media as defined in claim 9 wherein the steps performed by the programs further comprise clearing each of the read/write variables.
12. A computer system comprising:  
a first processor coupled to a first memory, the first processor and first memory forming a first functional unit;  
a second processor coupled to a second memory and forming a second functional unit, the second processor coupled to the first processor;  
a page table in the first functional unit having a virtual memory address (VMA) for a read/write variable, the VMA in the page table of the first functional unit pointing to the first memory; and  
a second page table in the second functional unit having a VMA for the read/write variable, the VMA in the page table of the second functional unit pointing to the second memory.
13. The computer system as defined in claim 12 further comprising:  
a first replicated program executing on the first processor, the first replicated program writing the read/write variable at a location indicated by the VMA in the page table of the first functional unit;

a second replicated program executing on the second processor, the second replicated program writing the read/write variable at a location indicated by the VMA in the page table of the second functional unit; and

wherein the first and second replicated programs are the copies of a same program.

14. The computer system as defined in claim 13 wherein the first and second replicated programs are copies of an operating system program, and wherein the read/write variable is a counter that indicates a number of executions of a code path of the operating system program.

15. The computer system as defined in claim 13 wherein the first and second replicated programs are copies of an operating system program, and wherein the read/write variable is a look-aside list header for process control blocks.

16. A computer system comprising:

a first means for executing programs coupled to a first means for storing programs and data, the first means for executing and first means for storing forming a first functional unit;

a second means for executing programs coupled to a second means for storing programs and data, and forming a second functional unit, the second means for executing coupled to the first means for executing;

a page table in the first functional unit having a virtual memory address (VMA) for a read/write variable, the VMA in the page table of the first functional unit pointing to the first means storing; and

a second page table in the second functional unit having a VMA for the read/write variable, the VMA in the page table of the second functional unit pointing to the second means for storing.

17. The computer system as defined in claim 16 further comprising:
  - a first replicated program executing on the first means for executing, the first replicated program writing the read/write variable at a location indicated by the VMA in the page table of the first functional unit;
  - a second replicated program executing on the second means for executing, the second replicated program writing the read/write variable at a location indicated by the VMA in the page table of the second functional unit; and

wherein the first and second replicated programs are the copies of a same program.
18. The computer system as defined in claim 17 wherein the first and second replicated programs are copies of an operating system program, and wherein the read/write variable is a counter that indicates a number of executions of a code path of the operating system program.
19. The computer system as defined in claim 17 wherein the first and second replicated programs are copies of an operating system program, and wherein the read/write variable is a look-aside list header for process control blocks.